Enabling Massive Scientific Datasets through Automated Schema Design

Year One Progress Report

Jeffrey P. Gardner, PI
Pittsburgh Supercomputing Center

Anastassia Ailamaki, co-PI
Carnegie Mellon University

Andrew Connolly, co-PI
University of Washington

1 Accomplishments

The first year of our project has faced a challenge because Connolly is moving from the University of Pittsburgh to the University of Washington (UW). Since none of the graduate students that he currently advises are moving with him, we felt that it was best if we postponed the start of the Pitt subcontract, transferred it to the UW, then started it with a UW graduate student once he got there. This has resulting in us delaying the start of the portion of the work that was allocated in to the University of Pittsburgh in the original proposal. Fortunately, the Connolly group's contributions are most useful during the second and third years of the project, so the loss of them during the first year is easiest to work around. The entire Pitt subcontract amount is being transferred to UW. Connolly's group will operate at an increased level of effort for the remaining two years of the project such that their total contribution will be the same as set forth in the original proposal.

In the meantime, we are pleased to report that much work as been done by the Ailamaki group, working closely with Gardner. Gardner conducted a cosmology simulation on the NCSA Teragrid cluster and ran common post-processing tools on it, specifically “group finders” that computational cosmologists typically use to identify interesting objects within the simulation such as galaxies and galaxy clusters. The parameters of the run were tuned for this project (frequent outputs, multiple species of particles, lots of galaxies and galaxy clusters that merge together). The Ailamaki group was able to assimilate the outputs of the simulation and the group finders into a database and implement our fiducial set of queries (currently 8 queries) on them. In accordance with our proposed project timeline, we are using the existing Autopart to optimize for each of these queries individually.

The regime where database technology can make the greatest impact is tracking group evolution over time. This is because simulation outputs are typically stored as snapshots. Thus, queries that focus on a single instant in time have an advantage over ones that examine time evolution simply because the file format naturally lends itself to the former kind of inquiry. We have found that scientists correspondingly limit their own research on this basis, preferring the “low lying fruit” of problems that can be addressed using single snapshot (or comparing at most two of them) and avoiding the ones that require looking at a range of snapshots. Our current query set, therefore, focuses on the time domain.

The most common type of query is "going back in time," i.e. identifying a set of progenitors for a given set of groups. Implementing this traversal in a database is not straightforward, requiring the development of "recursive" queries, which are not efficiently implemented in existing systems. The recursive aspect arises from the nature of the data. Tracking a group through sequential simulation snapshots is similar to traversing a doubly-linked list in memory. A given group *g* in output *t* knows its group ID in output *t*-1 and *t*+1. To follow the evolution of a group through many outputs, one must trace these links through many sequential outputs in the same manner that one traverses a linked list: e.g. galaxy 123 in output 10 is the same as galaxy 186 in output 9 which is the same as galaxy 452 in output 8, etc. Optimizing recursive query performance over large datasets is not well understood, and the tools currently available for automated indexing and partitioning do not work for recursive queries. We initially experimented with the obvious choice of flattening our database structures to eliminate recursive queries, but ended up with significantly increasing data sizes. Consequently, our current research focuses on optimizing the execution of recursive queries over large datasets and extending database indexing and partitioning tools to better support them.

2 Goals for next year

As an extension of our proposed strategy, we are dividing our learning query set into "computationalist" and "observer" queries. Our original proposal focused mainly on queries that a computational cosmologist would ask of the data from their own simulation. In the era of virtual observatories, an important contribution of massive simulations would be to have observers interact with them as well. Furthermore, we think that observers would make a natural user community of our technology since they are already used to interacting with databases in order to access observational survey data (e.g. the Sloan Digital Sky Survey provides an SQL interface for retrieving data products). Since one of the motivations of our project is that it enables computationalists to make their simulations readily accessible to the observational community, we should therefore include queries that observers are likely to make on simulated data. Connolly's group will serve as a source of "observer" queries, while Gardner will continue to represent the needs of computationalists.

Our goals for year two are to continue analyzing and optimizing Autopart for the computationalist queries. This is will be an iterative process. Our 8 fiducial queries are designed to be representative of a broad spectrum possible questions that an astronomer might be attempting to answer. But are these queries truly representative enough? After the first optimization stage, which will include optimizations for recursive queries, Gardner will take the existing technology and try using it to answer real scientific questions (e.g. "How many dwarf galaxies end up in clusters?" "How does merging history affect galaxy spin parameter?"). This process is meant to identify other queries that might be useful to add to our learning set. Starting upon Connolly's arrival at UW this summer, Connolly's group will provide a similar process for the "observer" learning set. By consistently focusing on the actual scientific process, we hope to iterate Autopart's partitioning algorithm towards a universally optimal strategy.